

AMENDMENTS

Please amend the application as follows:

In the Claims:

1. (Original) A system for enabling selective assertion testing of computer programs based on run time inputs, comprising:

memory; and

a compiler configured to translate a first function of a first program stored in said memory into a second function of a second program, said first function having assertion instructions, said second function having translated assertion instructions translated from said assertion instructions of said first function, said compiler configured to enable selective execution of a portion of said translated assertion instructions based on a run time input.
2. (Original) The system of claim 1, wherein said compiler is further configured to enable the execution of each said translated assertion instruction that extends an execution time of said second function to be prevented.
3. (Canceled)
4. (Original) The system of claim 1, wherein said compiler is further configured to detect stall locations in said second function and to insert into said stall locations each said translated assertion instruction that is outside of said portion.

5. (Original) The system of claim 4, wherein said compiler is further configured to enable execution of each said translated assertion instruction that is outside of said portion regardless of said run time input.

6. – 9. (Canceled)

7. (Original) A method for enabling selective assertion testing of computer programs based on run time inputs, comprising the steps of:

translating a first function of a first computer program into a second function of a second computer program, said first function having assertion instructions, said second function having translated assertion instructions translated from said assertion instructions of said first function;

detecting stall locations within said second function of said second computer program;

inserting one of said translated assertion instructions into one of said stall locations in response to a detection of said one stall location in said detecting step;

inserting a block of said translated assertion instructions into said second function; and

enabling selective execution of said block of translated assertion instructions based on a run time input.

8. (Original) The method of claim 7, further comprising the step of executing said one assertion instruction regardless of said run time input.

9. (Canceled)

10. (Original) The method of claim 7, further comprising the step of:
enabling execution of each said translated assertion instruction that is outside of
said block of translated assertion instructions regardless of said run time input.

11. (Original) A method for enabling selective assertion testing of computer
programs based on run time inputs, comprising the steps of:

translating a first function of a first computer program into a second function of
a second computer program, said first function having assertion instructions, said second
function having translated assertion instructions translated from said assertion
instructions of said first function; and

enabling selective execution of a portion of said translated assertion instructions
based on a run time input.

12. (Original) The method of claim 11, further comprising the step of
preventing execution of each said translated assertion instruction that extends an
execution time of said second function.

13. (Original) The method of claim 11, wherein said enabling step is
automatically performed along with said translating step.

14. (Original) The method of claim 11, further comprising the steps of:
detecting stall locations in said second function; and
inserting into said stall locations each said translated assertion instruction that is
outside of said portion.

15. (Original) The method of claim 14, further comprising the step of:
enabling execution of each said translated assertion instruction that is outside of said
portion regardless of said run time input.

19. (Canceled)

20. (New) A software testing system, comprising:
memory;
a compiler configured to translate source code into machine code, said source code
stored in said memory and comprising one or more assertion instructions and said machine code
stored in said memory and comprising one or more instructions representative of said source
code, the compiler further configured to determine stall locations that occur during execution of
said machine code and insert assertion instructions into said stall locations for execution, said
compiler further configured to insert mode test instructions into said machine code, said mode
test instructions configured to analyze a mode selection input and execute assertion handling
code based upon said mode selection input; and
a processor configured to request said mode selection input when said processor
executes said machine code.

21. (New) The software testing system of claim 20, wherein said mode selection
input indicates that the assertion handling code is to be executed upon detection by said mode
test instructions of a failure in said machine code.

22. (New) The software testing system of claim 21, wherein said mode test instructions are generated by said compiler.

23. (New) The software testing system of claim 21, wherein said mode test instructions are generated by a user.

24. (New) A software testing method, comprising:

translating source code into machine code, said source code comprising one or more assertion instructions and said machine code comprising one or more instructions representative of said source code;

determining stall locations that occur during execution of said machine code;

inserting assertion instructions into said stall locations for execution;

inserting mode test instructions into said machine code, said mode test instructions configured to analyze a mode selection input; and

analyzing said mode selection input based on said mode test instructions.

25. (New) The method of claim 24, further comprising the step of requesting said mode selection input from a user via a display device.

26. (New) The method of claim 25, further comprising the step of executing said assertion handling code based upon said mode selection input from said analyzing step.